Polar Codes: Robustness of the Successive Cancellation Decoder with Respect to Quantization

S. Hamed Hassani and Rüdiger Urbanke

***Abstract*—THIS PAPER IS ELIGIBLE FOR THE STUDENT PAPER AWARD. Polar codes provably achieve the capacity of a wide array of channels under successive decoding. This assumes infinite precision arithmetic. Given the successive nature of the decoding algorithm, one might worry about the sensitivity of the performance to the precision of the computation.**

We show that even very coarsely quantized decoding algorithms can lead to excellent performance. More concretely, we show that under successive decoding with an alphabet of cardinality only three, the decoder still has a threshold and this threshold is a sizable fraction of capacity. More generally, we show that if we are willing to transmit at a rate δ below capacity, then we need only $c\log(1/\delta)$ bits of precision, where c is a universal constant.

I. Introduction

Since the invention of polar codes by Arikan, [1], a large body of work has been done to investigate the pros and cons of polar codes in different practical scenarios (for a partial list see [2]-[8]).

We address one further aspect of polar codes using successive decoding. We ask whether such a coding scheme is *robust*. More precisely, the standard analysis of polar codes under successive decoding assumes infinite precision arithmetic. Given the successive nature of the decoder, one might worry how well such a scheme performs under a finite precision decoder. A priori it is not clear whether such a coding scheme still shows any threshold behavior and, even if it does, how the behavior scales in the number of bits of the decoder.

We show that in fact polar coding is extremely robust with respect to the quantization of the decoder. In Figure 1, we show the achievable rate using a simple successive decoder with only three messages, called the decoder with erasures, when transmission takes place over several important channel families. As one can see from this figure, in particular for channels with high capacity, the fraction of the capacity that is achieved by this simple decoder is close to 1, i.e., even this extremely simple decoder almost achieves capacity. We further show that, more generally, if we want to achieve a rate δ below capacity ($\delta > 0$), then we need at most[1] $c\log(1/\delta)$ bits of precision.

The significance of our observations goes beyond the pure computational complexity which is required. The main bottleneck in the implementation of large high speed coding systems is typically memory. Therefore, if one can find decoders which work with only a few bits per message then this can make the difference whether a coding scheme is implementable.

EPFL, School of Computer & Communication Sciences, Lausanne, CH-1015, Switzerland, {seyedhamed.hassani, rudiger.urbanke}@epfl.ch.

[1]All the logarithms in this paper are in base 2.

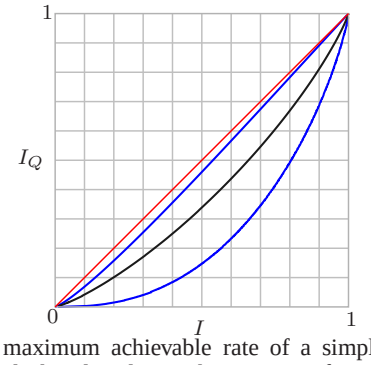

Fig. 1. The maximum achievable rate of a simple three message decoder, called the decoder with erasures, for different channel families. From top to bottom: the first curve corresponds to the family of binary erasure channels (BEC) where the decoder with erasures is equivalent to the original SC decoder and, hence, the maximum achievable rate is the capacity itself. The second curve corresponds to the family of binary symmetric channels (BSC). The third curve corresponds to the family of binary additive white Gaussian channels (BAWGN). The curve at the bottom corresponds to a universal lower bound on the achievable rate by the decoder with erasures.

A. Basic setting and definitions

Let $W : \mathcal{X} \to \mathcal{Y}$ be a binary memoryless symmetric (BMS) channel, with input alphabet $\mathcal{X} = \{0, 1\}$, output alphabet $\mathcal{Y}$, and the transition probabilities $\{W(y \,|\, x) : x \in \mathcal{X}, y \in \mathcal{Y}\}$. Also, let $I(W)$ denote the capacity of W.

Let $G_2 = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$. The generator matrix of polar codes is defined through the Kronecker powers of G_2, denoted by $G_N = G_2^{\otimes n}$. Throughout the paper, the variables N and n are related as $N = 2^n$. Let us quickly review how the generator matrix of polar codes is constructed. Consider the $N \times N$ matrix G_N and let us label the rows of the matrix G_N from top to bottom by $0, 1, \cdots, N-1$. Now assume that we desire to transmit binary data over the channel W at rate $R < I(W)$ with block-length N. One way to accomplish this is to choose a subset $\mathcal{I} \subseteq \{0, \cdots, N-1\}$ of size NR and to construct a vector $U_0^{N-1} = (U_0, \cdots, U_{N-1})$ in a way that it contains our NR bits of data at positions in $\mathcal{I}$ and contains, at positions not in $\mathcal{I}$, some fixed value (for example 0) which is known to both the encoder and decoder. We then send the codeword $X_0^{N-1} = U_0^{N-1} G_N$ through the channel W. We refer to the set $\mathcal{I}$ as the set of *chosen indices* or *information indices* and the set $\mathcal{I}^c$ is called the set of *frozen indices*. We explain in Section II-A how the good indices are chosen. At the decoder, the bits $u_0, \cdots, u_{N-1}$ are decoded one by one. That is, the bit u_i is decoded after $u_0, \cdots u_{i-1}$. If i is a frozen index, its value is known to the decoder. If not, the decoder estimate the value of u_i by using the output y_0^{N-1} and the estimates

of $u_0, \cdots, u_{i-1}$.

B. Quantized SC Decoder

Let $\mathbb{R}^* = \mathbb{R} \cup \{\pm\infty\}$ and consider a function $Q(x) : \mathbb{R}^* \to \mathbb{R}^*$ that is *symmetric* (i.e., $Q(x) = Q(-x)$). We define the Q-quantized SC decoder as a version of SC decoder in which the function Q is applied to the output of any computation that the SC decoder does. We denote such a decoder by SCD_Q.

Typically, the purpose of the function Q is to model the case where we only have finite precision in our computations perhaps due to limited available memory or due to other hardware limitations. Hence, the computations are correct within a certain level of accuracy which the function Q models. Thus, let us assume that the range of Q is a finite set $\mathcal{Q}$ with cardinality $|\mathcal{Q}|$. As a result, all the messages passed through the decoder SCD_Q belong to the set $\mathcal{Q}$.

In this paper we consider a simple choice of the function Q that is specified by two parameters: The distance between levels Δ, and truncation threshold M. Given a specific choice of M and Δ, we define Q as follows:

$$Q(x) = \begin{cases} \lfloor \frac{x}{\Delta} + \frac{1}{2} \rfloor \Delta, & x \in (0, M], \\ \lceil \frac{x}{\Delta} - \frac{1}{2} \rceil \Delta, & x \in [-M, 0), \\ \text{sign}(x) M, & \text{otherwise.} \end{cases} \tag{1}$$

Note here that $|\mathcal{Q}| = 1 + \frac{2M}{\Delta}$.

C. Summary of results

Theorem 1 (Main Statement): Consider transmission over a BMS channel W using polar codes and a SCD_Q with message alphabet $\mathcal{Q}$.

- For $|\mathcal{Q}| = 3$, we provide methods to precisely compute the maximum rate that can be achieved reliably when the transmission takes place over W and we use polar codes with the decoding algorithm SCD_Q. In particular, such maximum rates are plotted for different channel families in Figure 1. Also, in Figure 1 a universal lower bound for the maximum achievable rate is given. The methods used here are extendable to other quantized decoders.
- We can achieve up to an additive gap δ, $\delta > 0$, below the capacity $I(W)$ with $\log |\mathcal{Q}| \leq c \log(1/\delta)$.

Discussion: In short, polar codes are very robust to quantization within the decoder. In particular for BMS channels with capacity close to 1, very little is lost by quantizing.

The rest of the paper is devoted to proving Theorem 1. For the sake of briefness, we have omitted the proof of the lemmas stated in the sequel and we refer the reader to [10] for more details.

II. General Framework for Analysis

A. Equivalent tree channel model and analysis of the probability of error for the original SC decoder

Since we are dealing with a linear code, a symmetric channel and symmetric decoders throughout this paper, without loss of generality we confine ourselves to the *all-zero codeword* (i.e., we assume that all the u_i's are equal to 0). In order to better visualize the decoding process, the following definition is handy.

Definition 2 (Tree Channels of Height n): For each $i \in \{0, 1, \cdots, N-1\}$, we introduce the notion of the i-th tree channel of height n which is denoted by $T(i)$. Let $b_1 \dots b_n$ be the n-bit binary expansion of i. E.g., we have for $n = 3$, $0 = 000$, $1 = 001$, ..., $7 = 111$. With a slight abuse of notation we use i and $b_1 \cdots b_n$ interchangeably. Note that for our purpose it is slightly more convenient to denote the least (most) significant bit as b_n (b_1). Each tree channel consists of $n+1$ levels, namely $0, \dots, n$. It is a complete binary tree. The root is at level n. At level j we have 2^{n-j} nodes. For $1 \leq j \leq n$, if $b_j = 0$ then all nodes on level j are check nodes; if $b_j = 1$ then all nodes on level j are variable nodes. Finally, we give a label for each node in the tree $T(i)$: For each level j, we label the 2^{n-j} nodes at this level respectively from left to right by $(j, 0), (j, 1), \cdots, (j, 2^{n-j} - 1)$.

All nodes at level 0 correspond to independent observations of the output of the channel W, assuming that the input is 0.

An example for $T(011)$ (that is $n = 3$, $b = 011$ and $i = 3$) is shown in Fig. 2.

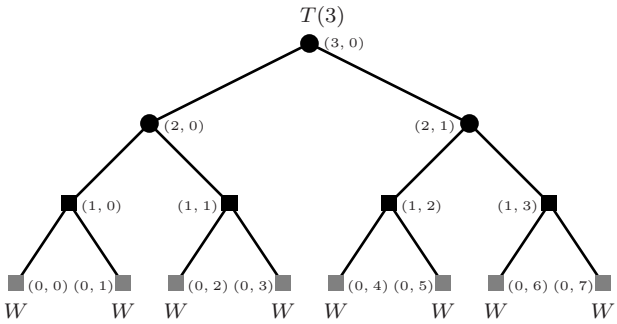

Fig. 2. Tree representation of the tree-channel $T(3)$. The 3-bit binary expansion of 3 is $b_1 b_2 b_3 = 011$ (note that b_1 is the most significant bit). The pair beside each node is the label assigned to it.

Given the channel output vector y_0^{N-1} and assuming that the values of the bits prior to u_i are given, i.e., $u_0 = 0, \cdots, u_{i-1} = 0$, we now compute the probabilities $p(y_0^{N-1}, u_0^{i-1} \,|\, u_i = 0)$ and $p(y_0^{N-1}, u_0^{i-1} \,|\, u_i = 1)$ via a simple message passing procedure on the equivalent tree channel $T(i)$. We attach to each node in $T(i)$ with label (j, k) a message[2] $m_{j,k}$ and we update the messages as we go up towards the root node. We start with initializing the messages at the leaf nodes of $T(i)$. For this purpose, it is convenient to represent the channel in the log-likelihood domain; i.e., for the node with label $(0, k)$ at the bottom of the tree which corresponds to an independent realization of W, we plug in the log-likelihood ratio (llr) $\log(\frac{W(y_k \,|\, 0)}{W(y_k \,|\, 1)})$ as the initial message $m_{0,k}$. That is,

$$m_{0,k} = \log(\frac{W(y_k \,|\, 0)}{W(y_k \,|\, 1)}). \tag{2}$$

Next, the SC decoder recursively computes the messages (llr's) at each level via the following operations: If the nodes at level j are variable nodes (i.e., $b_j = 1$), we have

[2]To simplify notation, we drop the dependency of the messages $m_{j,k}$ to the position i whenever it is clear from the context.

$$m_{j,k} = m_{j-1,2k} + m_{j-1,2k+1}, \tag{3}$$

and if the nodes at level j are check nodes (i.e., $b_j = 0$), the message that is passed up is

$$m_{j,k} = 2\tanh^{-1}(\tanh(\frac{m_{j-1,2k}}{2})\tanh(\frac{m_{j-1,2k+1}}{2})). \tag{4}$$

In this way, it can be shown that ([1]) the message that we obtain at the root node is precisely the value

$$m_{n,0} = \log(\frac{p(y_0^{N-1}, u_0^{i-1} \mid u_i = 0)}{p(y_0^{N-1}, u_0^{i-1} \mid u_i = 1)}). \tag{5}$$

Now, given (y_0^{N-1}, u_0^{i-1}), the value of u_i is estimated as follows. If $m_{n,0} = 0$ and we let $u_i = 0$. If $m_{n,0} < 0$ we let $u_i = 1$. Finally, if $m_{n,0} = 0$ we choose the value of u_i to be either 0 or 1 with probability $\frac{1}{2}$. Thus, denoting E_i as the event that we make an error on the i-th bit within the above setting, we obtain

$$\Pr(E_i) = \Pr(m_{n,0} < 0) + \frac{1}{2}\Pr(m_{n,0} = 0). \tag{6}$$

Given the description of $m_{n,0}$ in terms of a tree channel, it is now clear that we can use density evolution [9] to compute the the probability density function of $m_{n,0}$. In this regard, at each level j, the random variables $m_{j,k}$ are i.i.d. for $k \in \{0, 1, \cdots, 2^{n-j} - 1\}$. The distribution of the leaf messages $m_{0,k}$ is the distribution of the variable $\log(\frac{W(Y \mid 0)}{W(Y \mid 1)})$, where $Y \sim W(y \mid 0)$. One can recursively compute the distribution of $m_{j,k}$ in terms of the distribution of $m_{j-1,2k}, m_{j-1,2k+1}$ and the type of the nodes at level j (variable or check) by using the relations (3), (4) with the fact that the random variables $m_{j-1,2k}$ and $m_{j-1,2k+1}$ are i.i.d.

B. Quantized density evolution

An important point to note here is that with the decoder SCD_Q, the distribution of the messages in the trees $T(i)$ is different than the corresponding ones that result from the original SC decoder. Hence, the choice of the information indices is also specified by the choice of the function Q as well as the channel W.

For each label (j, k) in $T(i)$, let $\hat{m}_{j,k}$ represent the messages at this label. The messages $\hat{m}_{j,k}$ take their values in the discrete set $\mathcal{Q}$ (range of the function Q). It is now easy to see that for the decoder SCD_Q the messages evolve via the following relations. At the leaf nodes of the tree we plug in the message $\hat{m}_{0,k} = Q(\log(\frac{W(y_k \mid 0)}{W(y_k \mid 1)}))$, and the update equation for $\hat{m}_{(j,k)}$ is

$$\hat{m}_{j,k} = Q(\hat{m}_{j-1,2k} + \hat{m}_{j-1,2k+1}), \tag{7}$$

if the node (j, k) is a variable node and

$$\hat{m}_{j,k} = Q(2\tanh^{-1}(\tanh(\frac{\hat{m}_{j-1,2k}}{2})\tanh(\frac{\hat{m}_{j-1,2k+1}}{2}))), \tag{8}$$

if the node (j, k) is a check node. One can use the density evolution procedure to recursively obtain the densities of the messages $\hat{m}_{j,k}$.

Finally, let $\hat{E}_i$ denote the event that we make an error in decoding the i-th bit, with a further assumption that we have correctly decoded the previous bits $u_0, \cdots, u_{i-1}$. In a similar way as in the analysis of the original SC decoder, we get

$$\Pr(\hat{E}_i) = \Pr(\hat{m}_{n,0} < 0) + \frac{1}{2}\Pr(\hat{m}_{n,0} = 0). \tag{9}$$

Hence, one way to choose the information bits for the algorithm SCD_Q is to choose the bits u_i according to the least values of $\Pr(\hat{E}_i)$.

Note here that, since all of the densities takes their value in the finite alphabet $\mathcal{Q}$, the construction of such polar codes can be efficiently done in time $O(\mid \mathcal{Q} \mid^2 N \log N)$. We refer the reader to [1] to see how such a construction can be done.

C. Gallager Algorithm

Since our aim is to show that polar codes under successive decoding are robust against quantization, let us investigate an extreme case. The perhaps simplest message-passing type decoder one can envision is the Gallager algorithm. It works with single-bit messages. Does this simple decoder have a non-zero threshold? Unfortunately it does not, and this is easy to see.

We start with the equivalent tree-channel model. For each channel i of the polar code we have such a tree of height n and on each layer, nodes are either all check or all variable nodes. Since messages are only a single bit, the "state" of the decoder at level j can be described by a single non-negative number, namely the probability that the message at level j is incorrect. Assume that we transmit over a BSC(p). Let $x_0 = p \in (0, \frac{1}{2})$. We are interested in the evolution of x_j. This evolution depends of course on the sequence of levels, i.e., it depends on which tree channel we are considering.

Assume that x_j is given and that the next level consists of check nodes. In this case the error probability increases. More precisely, $x_{j+1} = 2x_j(1 - x_j) > x_j$ when $x_j \in (0, \frac{1}{2})$. In other words, the state deteriorates. What happens if the next level consists of variable nodes instead? A little thought shows that in this case $x_{j+1} = x_j$, i.e., there is no change at all. This is true since if both incoming messages agree we can make a decision on the outgoing message, but if they differ we can only guess. This gives us $x_{j+1} = x_j^2 + x_j(1 - x_j) = x_j$.

Since in either case, the state either becomes worse or stays unchanged, no progress in the decoding is achieved, irrespective of the given tree. In other words, this decoder has a threshold of zero. As we have seen, the problem is the processing at the variable nodes since no progress is achieved there. But since we only have two incoming messages there is not much degree of freedom in the processing rules.

D. 1-Bit Decoder with Erasures

Motivated by the previous example, let us now add one message to the alphabet of the Gallager decoder, i.e., we also add the possibility of having erasures to the above mentioned Gallager algorithm. In this regard, function $Q(x)$ becomes the

sign function[3], i.e.,

$$Q(x) = \begin{cases} \infty & x > 0, \\ 0 & x = 0, \\ -\infty & x < 0. \end{cases} \tag{10}$$

As a result, all messages passed by the algorithm SCD_Q take on only three possible values: $\{-\infty, 0, \infty\}$. In this regard, the decoding procedure takes a very simple form. The algorithm starts by quantizing the channel output to one of the three values in the set $\mathcal{Q} = \{-\infty, 0, \infty\}$. At a check node we take the product of the signs of the incoming messages and at a variable node we have the natural addition rule ($0 \leftarrow \infty + -\infty$, $0 \leftarrow 0 + 0$ and $\infty \leftarrow \infty + \infty, \infty \leftarrow \infty + 0$ and $-\infty \leftarrow -\infty + -\infty, -\infty \leftarrow -\infty + 0$). Note that on the binary erasure channel, this algorithm is equivalent to the original SC decoder.

We now compute the maximum possible rate that the decoder SCD_Q can achieve reliably for a BMS channel W (we denote it by $C(W, Q)$). The analysis is done in three steps:

1) The density evolution procedure: To analyze the performance of this algorithm, first note that since all our messages take their values in the set $\mathcal{Q}$, then all the random variables that we consider have the following form

$$D = \begin{cases} \infty & \text{w.p. } p, \\ 0 & \text{w.p. } e, \\ -\infty & \text{w.p. } m. \end{cases} \tag{11}$$

Here, the numbers p, e, m are probability values and $p + e + m = 1$. Let us now see how the density evolves through the tree-channels. For this purpose, one should trace the output distribution of (7) and (8) when the input messages are two i.i.d. copies of a r.v. D with pdf as in (11).

Lemma 3: Given two i.i.d. versions of a r.v. D with distribution as in (11), the output of a variable node operation (7), denoted by D^+, has the following form

$$D^+ = \begin{cases} \infty & \text{w.p. } p + 2pe, \\ 0 & \text{w.p. } e^2 + 2pm, \\ -\infty & \text{w.p. } m^2 + 2em. \end{cases} \tag{12}$$

Also, the check operation (8), yields D^- as

$$D^- = \begin{cases} \infty & \text{w.p. } p^2 + m^2, \\ 0 & \text{w.p. } 1 - (1-e)^2, \\ -\infty & \text{w.p. } 2pm. \end{cases} \tag{13}$$

In order to compute the distribution of the messages $\hat{m}_{n,0}$ at a given level n, we use the method of [1] and define the polarization process D_n as follows. Consider the random variable $L(Y) = \log(\frac{W(Y\,|\,0)}{W(Y\,|\,1)})$, where $Y \sim W(y\,|\,0)$. The stochastic process D_n starts from the r.v. $D_0 = Q(L(Y))$ defined as

$$D_0 = \begin{cases} \infty & \text{w.p. } p = \Pr(L(Y) > 0), \\ 0 & \text{w.p. } e = \Pr(L(Y) = 0), \\ -\infty & \text{w.p. } m = \Pr(L(Y) < 0). \end{cases} \tag{14}$$

and for $n \geq 0$

$$D_{n+1} = \begin{cases} D_n^+ & ;\text{w.p. } \frac{1}{2}, \\ D_n^- & ;\text{w.p. } \frac{1}{2}, \end{cases} \tag{15}$$

[3]Note here that to fit such a function $Q(x)$ to the definition (1), we have assumed that $M = \Delta$ and Δ is close to 0.

where the plus and minus operations are given in (12), (13).

2) Analysis of the process D_n: Note that the output of process D_n is a itself a random variable of the form given in (11). Hence, we can equivalently represent the process D_n with a triple (m_n, e_n, p_n), where the coupled processes m_n, e_n and p_n are evolved using the relations (12) and (13) and we always have $m_n + e_n + p_n = 1$. Following along the same lines as the analysis of the original SC decoder in [1], we first claim that as n grows large, the process D_n will become polarized, i.e., the output of the process D_n will almost surely be a completely noiseless or a completely erasure channel.

Lemma 4: The random sequence $\{D_n = (p_n, e_n, m_n), n \geq 0\}$ converges almost surely to a random variable D_∞ such that D_∞ takes its value in the set $\{(1, 0, 0), (0, 1, 0)\}$.

We now aim to compute the value of $C(W, Q) = \Pr(D_\infty = (1, 0, 0))$, i.e., the ratio of the noiseless indices. The value of $\Pr(D_\infty = (1, 0, 0))$ is dependent on the starting channel D_0 that is given in (14) and is the highest rate that we can achieve with the 1-bit decoder with erasures. Let us first note that a density D as in (11) can be equivalently represented as a simple BMS channel given in Fig. 3. This equivalence stems

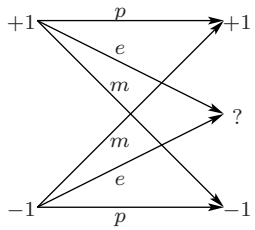

Fig. 3. The equivalent channel for the density D given in (11).

from the fact that for such a channel, conditioned on the event that the symbol $+1$ has been sent, the distribution of the output is precisely D. With a slight abuse of notation, we also denote the corresponding BMS channel by D. In particular, it is an easy exercise to show that the capacity ($I(D)$), Bhattacharyya parameter ($Z(D)$) and the error probability ($E(D)$) of the density D are given as

$$I(D) = (m + p)(1 - h_2(\frac{p}{p+m})), \tag{16}$$

$$Z(D) = 2\sqrt{mp} + e, \tag{17}$$

$$E(D) = 1 - p - \frac{e}{2}. \tag{18}$$

where $h_2(\cdot)$ denotes the binary entropy function. It is now clear that since the function Q is a not injective, we have

$$\frac{I(D^+) + I(D^-)}{2} \leq I(D).$$

This implies that the process $I_n = I(D_n)$ is a bounded supermartingale. Hence, I_n converges a.s. to a limit random variable I_∞. Furthermore, since $I(D = (1, 0, 0)) = 1$ and $I(D = (0, 1, 0)) = 0$, we deduce from Lemma 4 that I_∞ is a $0 - 1$ valued random variable and hence

$$C(W, Q) = \Pr(D_\infty = (1, 0, 0)) = \Pr(I_\infty = 1) = \mathbb{E}(I_\infty).$$

Now, from the fact that I_n is a supermartingale, we obtain a sequence of upper bounds on $C(W, Q)$ as follows. For $n \in \mathbb{N}$

we have

$$C(W,Q) \leq \mathbb{E}[I_n]. \tag{19}$$

In a similar way, one can obtain a sequence of lower bounds for $C(W,Q)$.

Lemma 5: Define the function $F(D)$ as $F(D) = p - 4\sqrt{pm}$ for $D \in \mathcal{D}$. We have $F(D = (1,0,0)) = 1$, $F(D = (0,1,0)) = 0$ and

$$\frac{F(D^+) + F(D^-)}{2} \geq F(D). \tag{20}$$

Hence, the process $F_n = F(D_n)$ is a submartingale and the for $n \in \mathbb{N}$ we have

$$C(W,Q) \geq \mathbb{E}[F_n]. \tag{21}$$

Given a BMS channel W, one can numerically compute $C(W,Q)$ with arbitrary accuracy δ: Consider the two functions $I(D)$ and $F(D)$ defined above. The values $\mathbb{E}[I(D_n)]$ and $\mathbb{E}[F(D_n)]$ can be computed in time $O(2^n)$. Let $n \in \mathbb{N}$ be such that $\mathbb{E}[g(D_n)] - \mathbb{E}[h(D_n)] \leq \delta$. Since $C(W,Q)$ is sandwiched between the two, then $\mathbb{E}[h(D_n)]$ provides a lower bound on $C(W,Q)$ which is no further from it than δ. The curves in Figure 1 have been plotted with these considerations. Also, for a channel W with capacity $I(W)$ and error probability $E(W)$, we have

$$E(W) \leq \frac{1 - I(W)}{2}. \tag{22}$$

Therefore, $\inf_{\{D : E(D) = \frac{1-I(W)}{2}\}} C(D,Q) \leq C(W,Q)$, which leads to the universal lower bound obtained in Figure 1.

3) Scaling behavior and error exponent: In the last step, we need to show that for the rates below $C(W,Q)$ the block-error probability decays to 0 for large block-lengths.

Lemma 6: Let $D \in \mathcal{D}$. We have

$$Z(D^-) \leq 2Z(D) \text{ and } Z(D^+) \leq 2(Z(D))^{\frac{3}{2}}.$$

Hence, for transmission rate $R < C(W,Q)$ and block-length $N = 2^n$, the probability of error of SCD_Q, denoted by $P_{e,Q}(N,R)$ satisfies $P_{e,Q}(N,R) = o(2^{-N^\beta})$ for $\beta < \frac{\log \frac{3}{2}}{2}$.

E. Trade-off between the number of bits and the gap to capacity

In this section, we give a rough sketch of the proof of the second part of Theorem 1. Consider a BMS channel W and assume that we need an algorithm SCD_Q such that is capable of achieving rates up to $I(W) - d$, where $d \leq \frac{1}{2}$ is a positive constant (for $d \geq \frac{1}{2}$ the 1-bit decoder with erasures is already a good choice). We first consider the original SC decoder and choose an integer n_d large enough so that for $n \geq n_d$, at least a fraction $I(W) - \frac{d}{2}$ of the sub-channels at level n have Bhattacharyya value less than e^{-2n}. As a result, if we perform the original SC decoding, then at level n at least a fraction $I(W) - \frac{d}{2}$ of the sub-channels are very perfect. Let $\mathcal{I}_d$ denote the set of indices of these sub-channels. In the second step, we tune the parameters M and Δ for a decoder SCD_Q (with function Q given in (1)) in a way that the algorithm SCD_Q still decodes perfectly on the indices that belong to the set $\mathcal{I}_d$. In order to choose such suitable parameters, the following lemma is useful.

Lemma 7: Fix $n \in \mathbb{N}$ and let $M = 2n$ and also $\Delta = 2^{-(n+1)}$. Then with probability at least $1 - 16(n+2)(\frac{2}{e})^{2n}$, the decoder SCD_Q outputs the message $+\infty$ at each of the positions $i \in \mathcal{I}_d$.

Hence, if we fix M and Δ as in Lemma 7 then at each index $i \in \mathcal{I}_d$, with probability at most $16(n+2)(\frac{2}{e})^{2n}$ we get a message other than ∞. This implies that at $i \in \mathcal{I}_d$ the distribution of the messages that we get by the algorithm SCD_Q stochastically dominates the following distribution

$$D = \begin{cases} \infty & \text{w.p. } 1 - 16(n+2)(\frac{2}{e})^{2n}, \\ -\infty & \text{w.p. } 16(n+2)(\frac{2}{e})^{2n}. \end{cases} \tag{23}$$

In the third step, we provide a lower bound on the final ratio of the perfect sub-channels that are branched out from the indices in $\mathcal{I}_d$. For this, we use Lemma 5 and note (without proof) that this ratio is at least the maximum rate that the 1-bit decoder with erasures can achieve, which is

$$C(D,Q) \geq p - 4\sqrt{pm} \geq 1 - 16(n+2)(\frac{2}{e})^{2n} - 16\sqrt{n+2}(\frac{2}{e})^{n}. \tag{24}$$

In the last step, we put things together. We first choose n_1 so that for $n \geq n_1$ the lower bound in (24) is at least equal to $1 - \frac{d}{2}$. We then set $n = \max(n_d, n_1)$ and choose the values of M and Δ according to Lemma 7, i.e., $M = 2n$ and $\Delta = 2^{-(n+1)}$. For this choice of M and Δ, the range of function Q is roughly $|\mathcal{Q}| = 4n2^{n+1}$. This implies that we need $\log(Q) = \log n + n + 3$ bits of precision for the algorithm SCD_Q. Finally, we establish the relation between n and d by proving that $n \geq 7\log(\frac{1}{d}) + \log(\log(\frac{1}{d}))^2 + 48$. The reader is referred to [10] for a detailed explanation of this proof.

REFERENCES

[1] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Info. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[2] R. Mori and T. Tanaka, "Performance and construction of polar codes on symmetric binary-input memoryless channels," in *Proc. 2009 IEEE Int. Symp. Info. Theory*, Seoul, South Korea, pp. 1496–1500, 2009.

[3] I. Tal and A. Vardy, "How to construct polar codes," presented at 2010 IEEE Info. Theory Workshop, Dublin, Ireland, 2010. [online] Available: `arXiv:1105.6164v1 [cs.IT]`.

[4] C. Leroux, I. Tal, A. Vardy and W. J. Gross, "Hardware architectures for successive cancellation decoding of polar codes," in *Proc. ICASSP 2011*, Prague, Czech Republic, pp. 1665-1668, 2011.

[5] R. Pedarsani, H. Hassani, I. Tal and E. Telatar, "On the construction of polar codes," in *Proc. 2011 IEEE Int. Symp. Info. Theory*, St. Petersberg, Russia, pp. 11-15, 2011.

[6] I. Tal and A. Vardy, "List decoding of polar codes," in *Proc. 2011 IEEE Int. Symp. Info. Theory*, St. Petersberg, Russia, pp. 1-5, 2011.

[7] S. B. Korada, A. Montanari, E. Telatar and R. Urbanke, "An empirical scaling law for polar codes," in *Proc. 2010 IEEE Int. Symp. Info. Theory*, Austin, Texas, USA, pp. 884-888, 2010.

[8] S. H. Hassani, K. Alishahi and R. Urbanke, "On the scaling of polar codes: II. The behavior of un-polarized channels," in *Proc. 2010 IEEE Int. Symp. Info. Theory*, Austin, Texas, USA, pp. 879–883, 2010.

[9] T. Richardson and R. Urbanke, *Modern coding theory*. Cambridge University Press, 2008.

[10] S. H. Hassani and R. Urbanke, "Polar Codes: Robustness of the Successive Cancellation Decoder with Respect to Quantization," [online] Available: `https://infoscience.epfl.ch/record/174733`.